AUSTRALIAN JOURNAL OF ENGINEERING AND TECHNOLOGY RESEARCH (AJMFR)
Vol.1, Issue 1 - 2016

DESIGN OF MAINTENANCE MANAGEMENT SYSTEM FOR MANUFACTURING ORGANIZATIONS

Muhammad Nabeel Musharraf

ABSTRACT:

Maintenance is an extremely important function that ensures the availability and reliability of production systems. In order for maintenance to function effectively, an effective information management system is essential. However, we see that many commonly found maintenance management systems lack important functions. In this paper, we have presented the development process for a medium-sized maintenance information management system and reviewed various phases of its design, development, and testing including relevant functions and processes. The proposed model can be customized to adjust the scope and to be contextualized according to specific organizational needs.

Introduction:

Information systems have revolutionized the way industrial operations are managed nowadays (Fountas et al., 2015). Long gone are the days when the bill of material (BOM) for equipment used to be searched from hand-written or dot-matrix printers. It is now available in a single click through management information systems designed for various departments within and between the companies and organizations. Considering a vast number of equipment, spare parts, maintenance procedures and techniques used in the field of maintenance, the importance of MIS is unquestionable. In this paper, we will propose a design for an effective maintenance information management system.

Objective:

This paper aims to explain the key aspects related to design including:

- Design basis,
- Use cases,
- Entity relationship diagrams, and
- Test cases.

Project Significance:

Modern maintenance management is not to repair broken equipment rapidly. Modern maintenance management is to keep the equipment running at high capacity and produce quality products at the lowest cost possible. Maintenance information management systems play a huge role in this.

This project can be used as a framework for the design of maintenance management system in terms of breaking down the complex maintenance aspects to manageable components which can be further built upon and developed.

The proposed design ensures departmental integrations with and within Maintenance function and accordingly carries a huge potential for maintenance management optimization through this MIS.

Actors Catalogue:

An actor may perform many different functions or play many different roles in a system. Also, there may be many actors in the system.

For the purpose of this project, key actors considered in this system will be the following:

- Supply chain team
- Maintenance team
- Management
- Engineering Spares Team
- Production Team

Functional Requirements:

System requirements are a list of necessary functions, capabilities, or characteristics related to the system being developed and the plans for creating it. There are several types of requirements that may be defined during the process that comes together to focus and prioritize the project plan. System requirements can be of various types and 'functional requirements' is one of its these different types. Functional Requirements provide details of how a product should behave and specify what is needed for development ("Website Requirements," 2013).

Following will be the functional requirements for proposed design:

- Ability to store Master Data (including names and numbers of machines, spare parts BOM, maintenance BOM, Type of maintenance, budgets etc)
- Ability to produce and store 12 months (rolling) maintenance plan based on master data and functional inputs, retrievable by relevant functions
- Ability to store breakdown/ failure records and prepare necessary statistics in order to facilitate decision-making
- Ability to incorporate special maintenance requests under exceptional circumstances
- Ability to generate overall maintenance and machine-wise costs
- Ability to generate reports
- Ability to generate spares ordering schedule

Use Cases:

A use case is a written description of how users will perform tasks on your system. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled ("Use Cases," 2013). US government's 'usability' mentions: "Use cases add value because they help explain how the system should behave and in the process, they also help brainstorm what could go wrong. They provide a list of goals and this list can be used to establish the cost and complexity of the system. Project teams can then negotiate which functions become requirements and are built ("Use Cases," 2013).

Following is the broad overall use-case diagram for the project showing various interactions in line with the functional requirements:

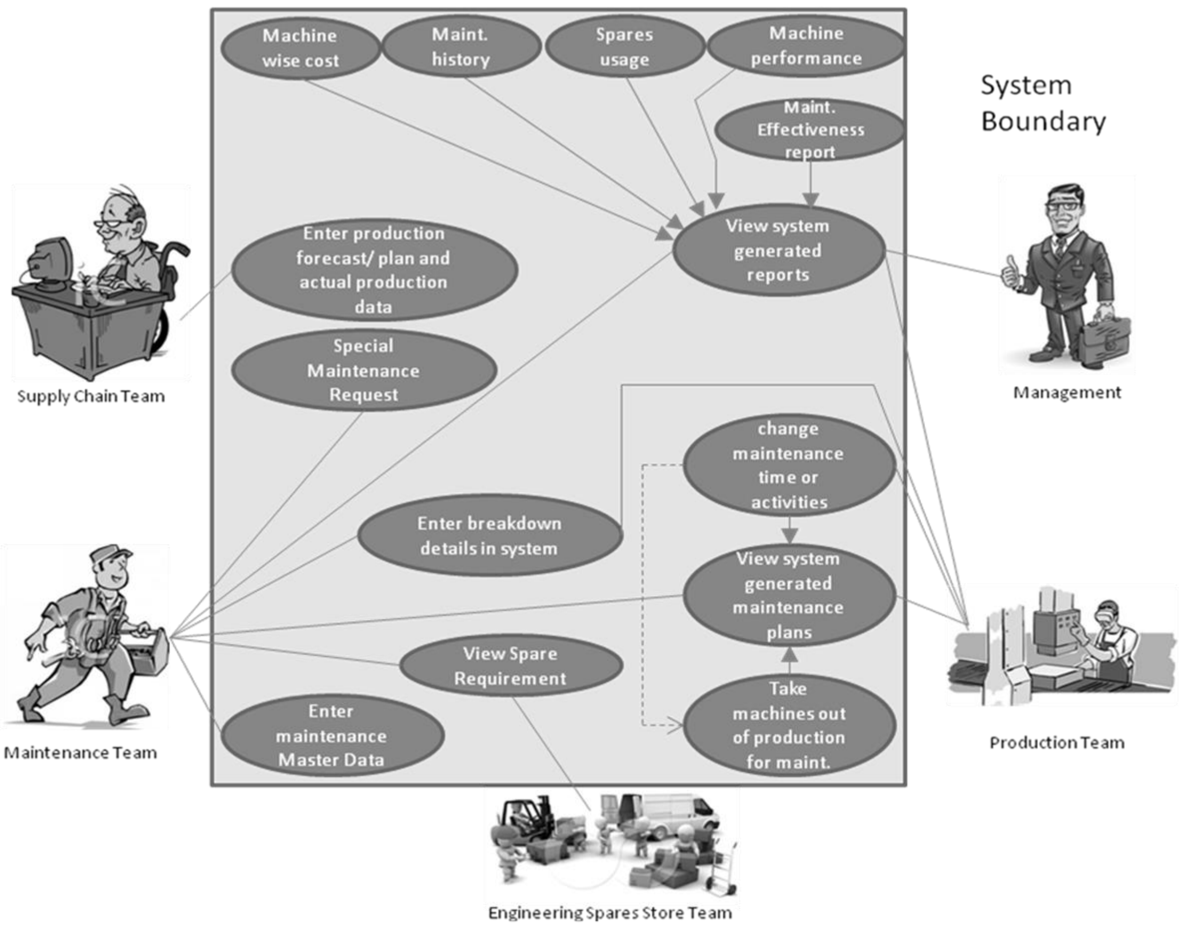

Figure 1: System Use Case

This broad use-case diagram has been divided into further individual cases to provide a better overview of relationships and interaction with systems.

Maintenance Team enters master-data in the system. This master-data is used for managing spares with regards to their life, replacement schedules, and maintenance plans:

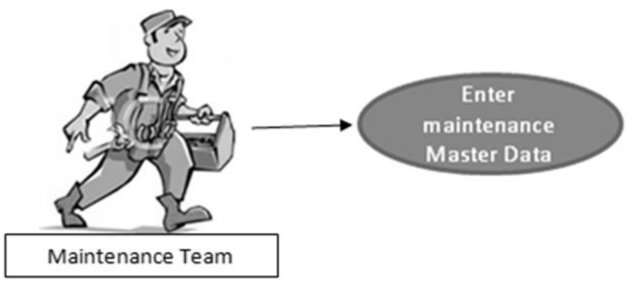

Figure 2: Entry of Master-data

Supply chain team enters production forecast, production plan, and actual performance. Maintenance of equipment would be dependent on these forecasts. Also, there would be some fixed life spares which are also classified as 'fast-movers'. Their availability in stores would be dependent on correct entry of production data. System would be able to generate the purchase orders for suppliers based on production forecast and actual performance in accordance with re-order levels:

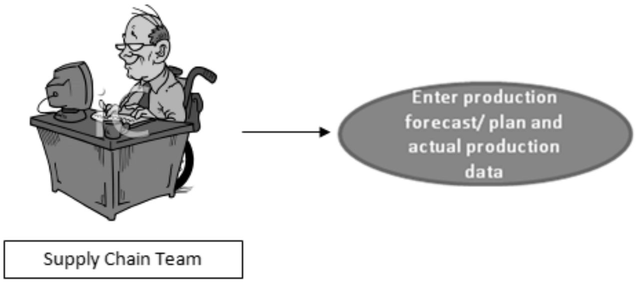

Figure 3: Entry of production data

Engineering team enters the spares requirement for various equipment in the system. Maintenance team extracts the requirements from system in case of planned maintenance activities and any specific spares required during pre-defined breakdown repairs:

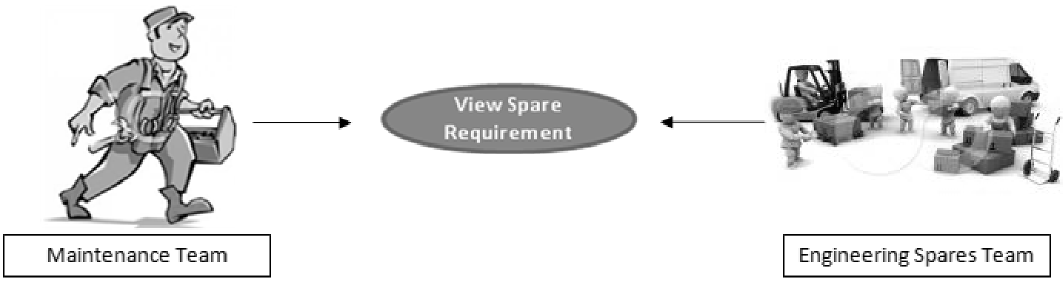

Figure 4: Entry of spare parts requirements

Maintenance teams and productions teams can raise work orders to any special maintenance requirement identified during operation or maintenance. It is important to record these events so that spares can be issued against particular equipment. This would help managements and other relevant stakeholders view how each piece of equipment is performing and if there are any trends. This information can have also other uses e.g. analysis of usage for a particular type of spare. System reports can show a rise in spare requirement on an equipment or across the board which can then be followed up by actions.

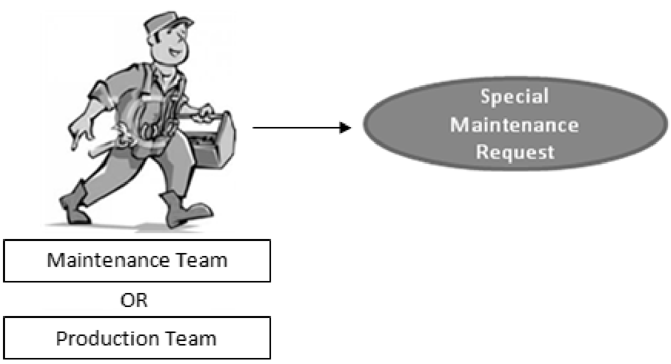

Figure 5: Entry of special maintenance requests

Production and maintenance teams need to be able to input the breakdowns. Its reasons justifications are the same as given in above use-case.

Figure 6: Entry of breakdown details

Equipment has to be maintained in line with relevant manufacturer guideline, organizational procedures, and other aspects. These plans should not be based on memory and should be fed in the system for an automatic generation at given timeframes. Maintenance and production teams should be able to view these plans so that they can plan their operations, rosters and other relevant aspects.

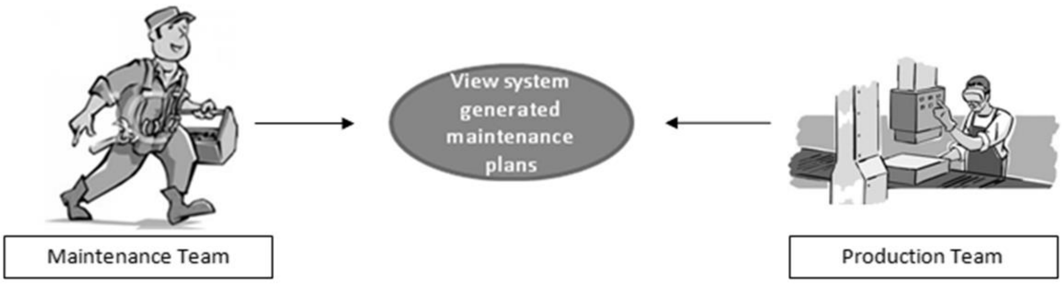

Figure 7: Entry of system generated maintenance plans

Maintenance plans may need to be changed due to factors such as unavailability of spares, results of predictive maintenance, operational and maintenance priorities etc. This capability should accordingly be there is the system. Also, it is important that machines are taken out of production in the system, irrespective if the maintenance is planned or unplanned, so that system can assess the capability to meet production demands. If demands cannot be met because of some unplanned maintenance work, the system can report this to relevant stakeholders.

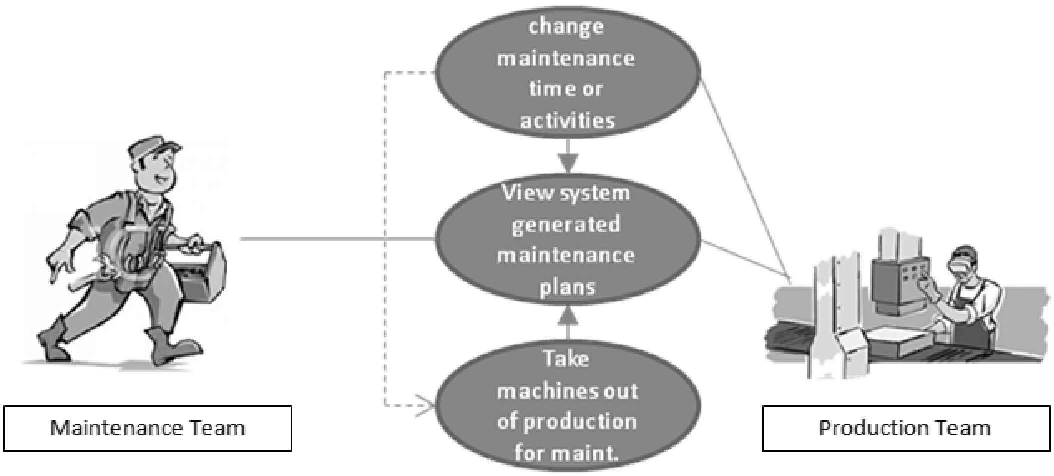

Figure 8: Operational activities during maintenance

Based on this information system can generate a number of reports for stakeholders.

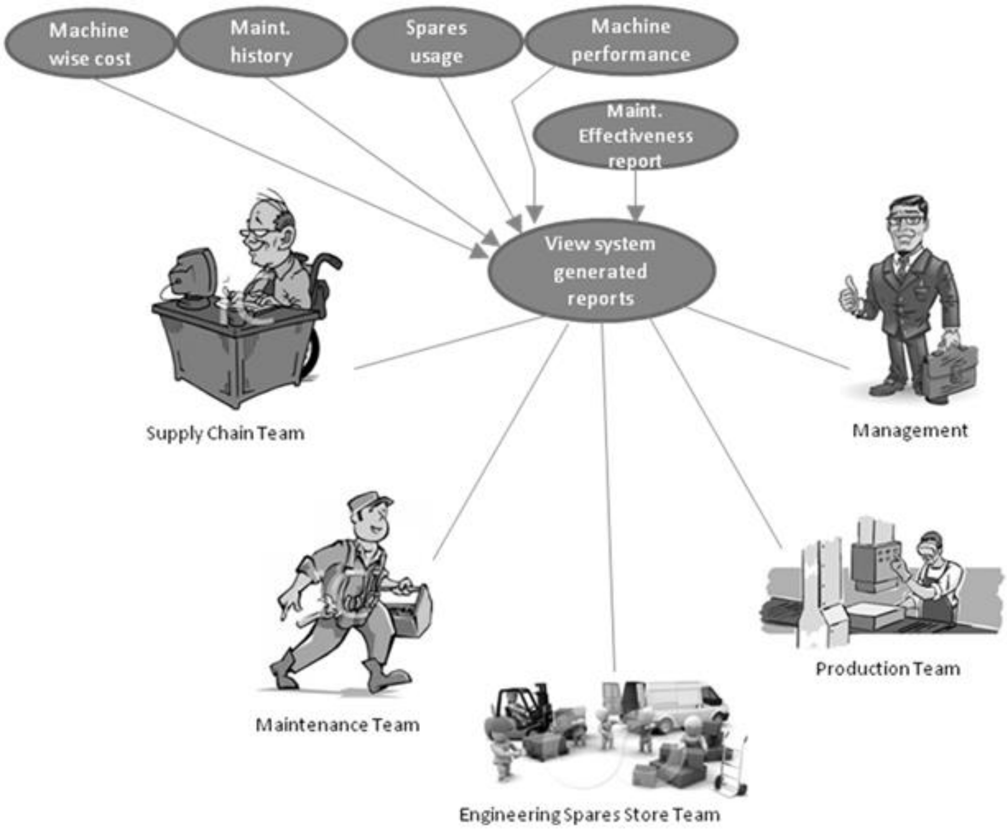

Figure 9: System generated reports

ARCHITECTURAL DESIGN:

System's architectural design is explained in figure 10.

The architecture consists of three layers which represent functions and relevant HMI of the system based on above mentioned use-cases.

Presentation layer:

The presentation layer provides the application's user interface (Microsoft Corporation, 2015). This layer in our proposed system architecture provides facility to various stakeholders to interact with the system as shown in figure 10 which also highlights relevant screens that are required to be built in the HMI (human machine interface).

Logic Layer:

In this layer, aspects which are required to be covered in the programming phase with regards to system functional requirements are explained.

Data Layer:

The data layer provides access to external systems such as databases. Figure 10 shows relevant information which is required to be stored.

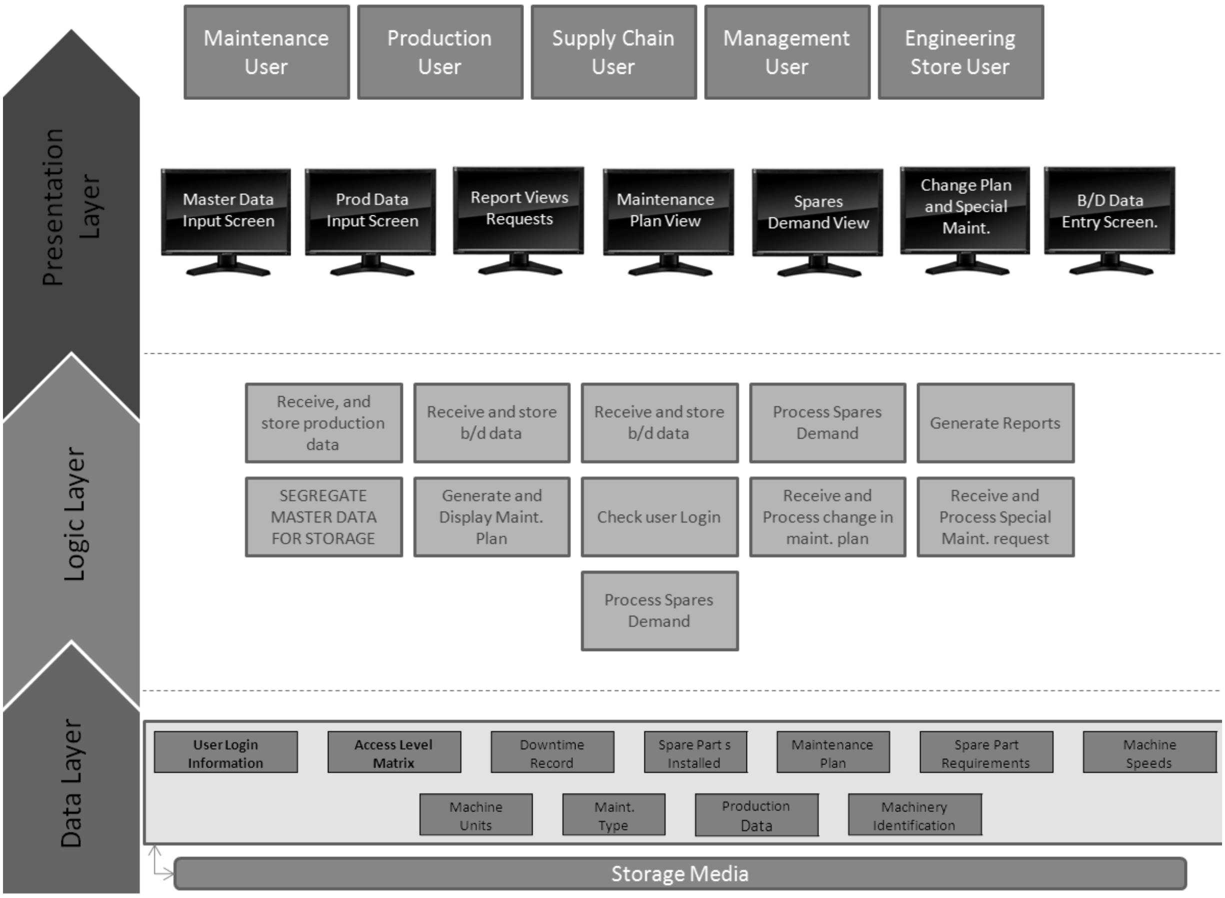

Figure 10: System Architecture

Hardware architecture is defined in three tiers:

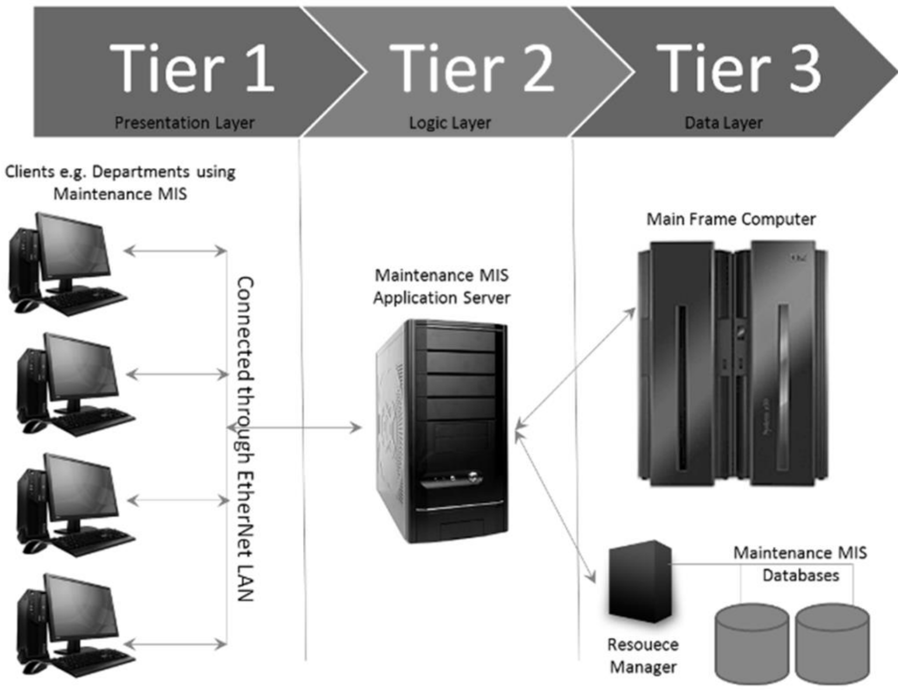

Figure 11: Hardware Architecture

PROJECT SCOPE:

Based on use-cases explained in previous sections, following is a description of proposed systems project scope and relevant interactions with various stakeholders.

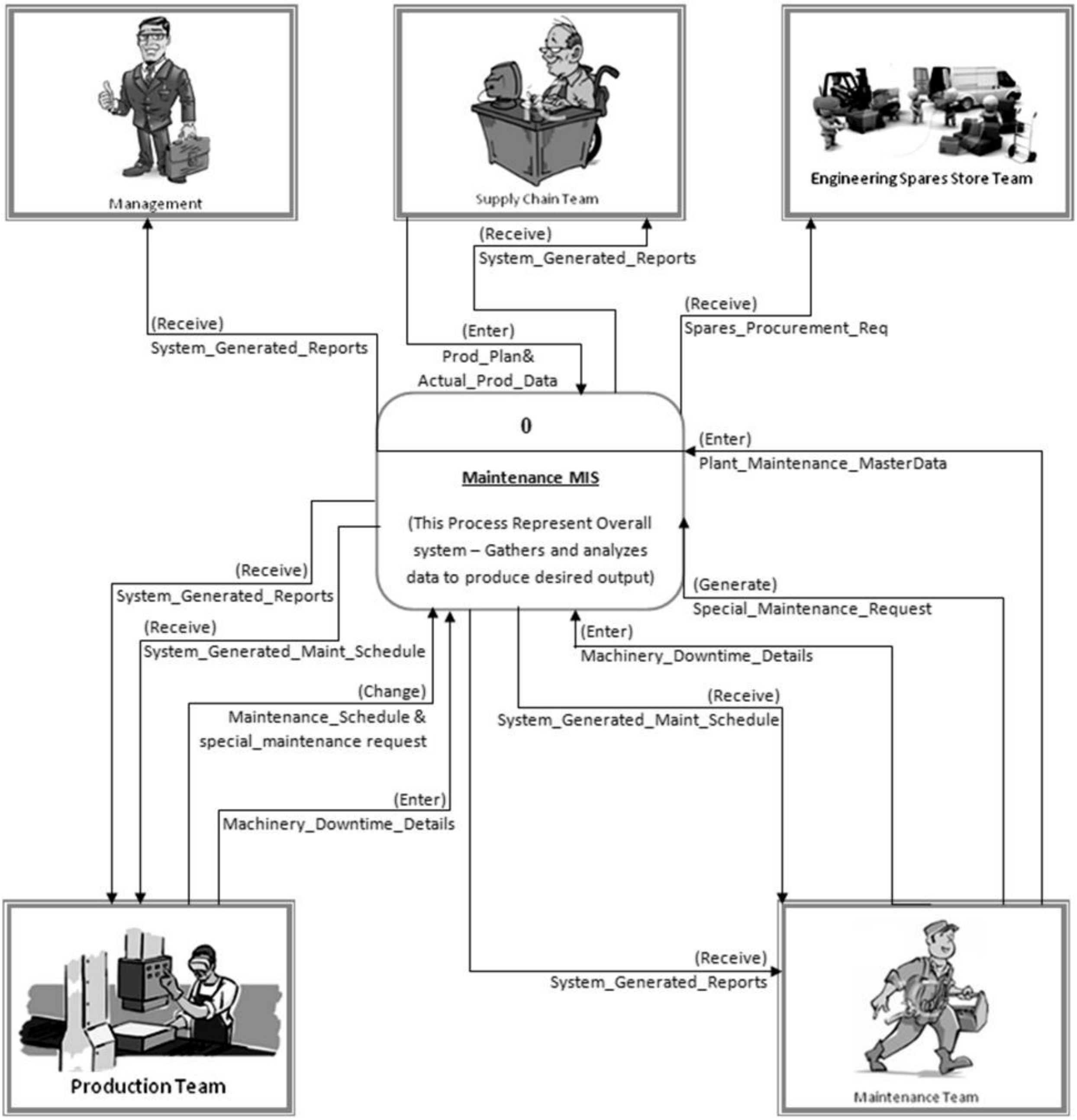

Figure 12: Project Scope

Above figure shows level zero data-flow diagram which considers the whole system as a single entity, like a black-box without any attention being paid to what is happening in that box.

Further details data-flow diagrams explore the inside of this box and explain the internal functionality.

DETAILED DATA-FLOW DIAGRAM:

Following is the detailed dataflow diagram for the system showing internal processes.

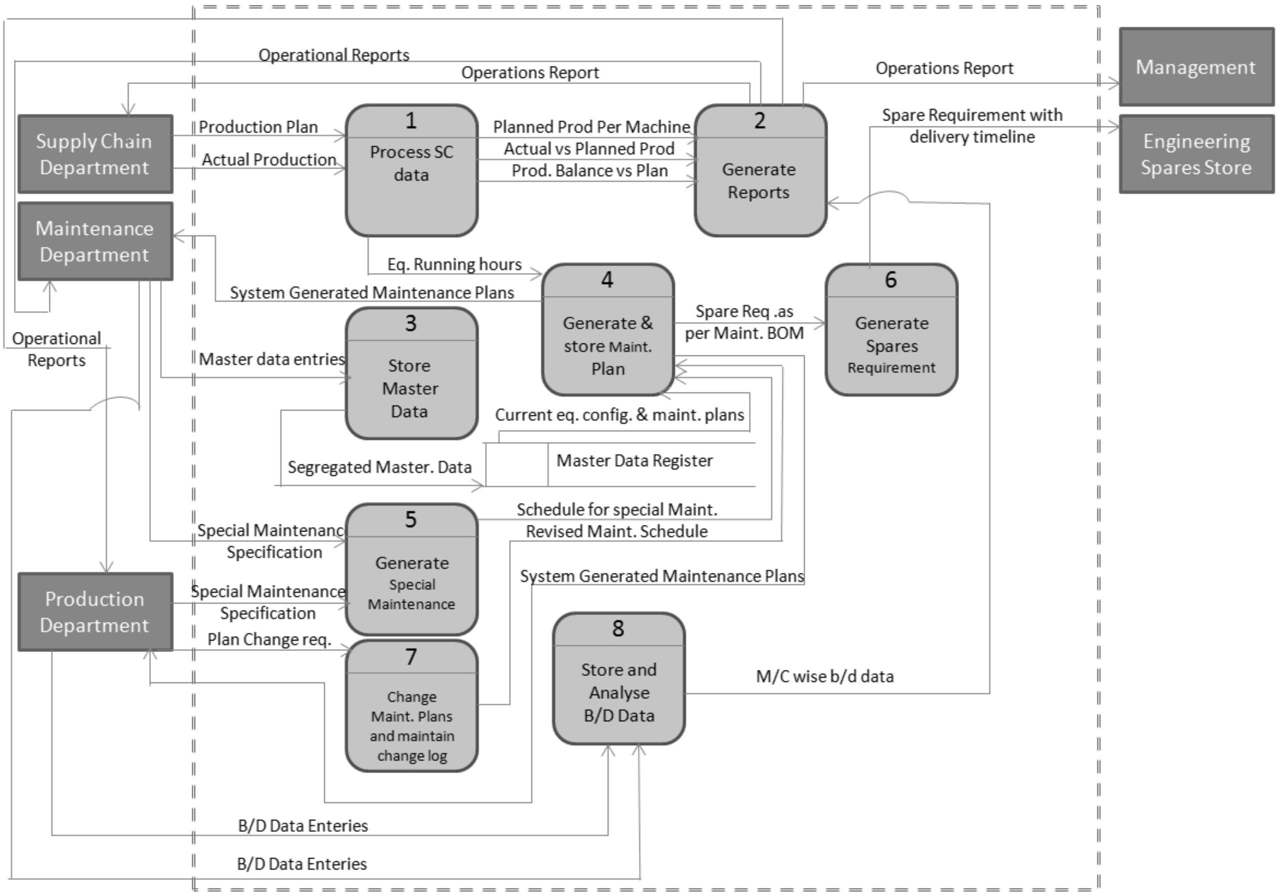

Figure 13: System Dataflow diagram

SEQUENCE DIAGRAMS:

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

With regards to the current project, we have prepared detailed sequence diagrams explaining various internal functions, data movement, storage and stakeholders for this system.

Sequence diagrams have been presented below.

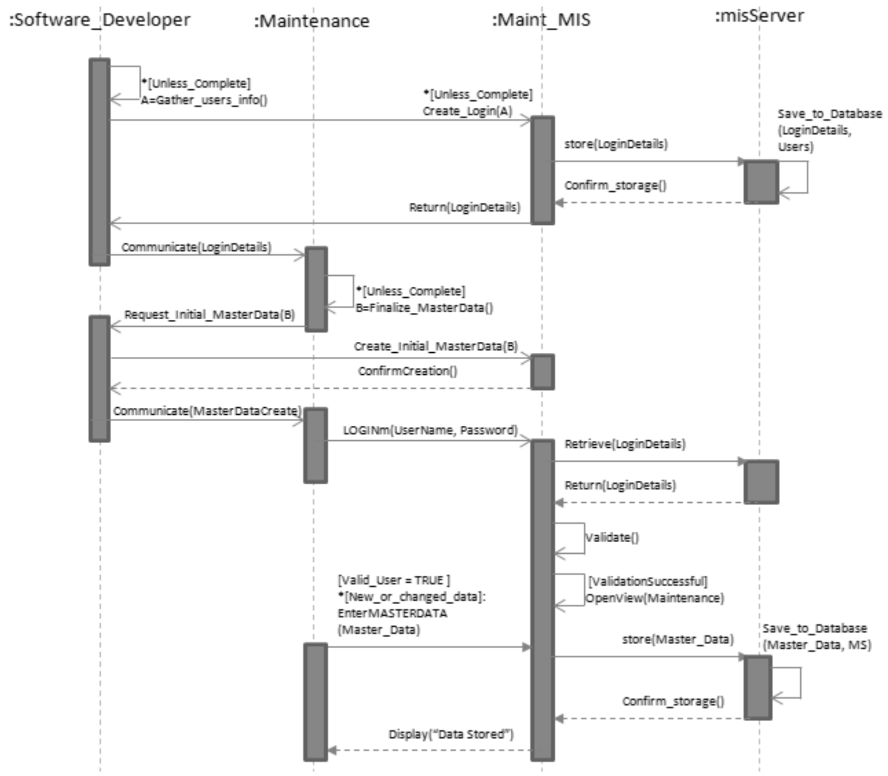

Figure 14. Sequence diagram 1

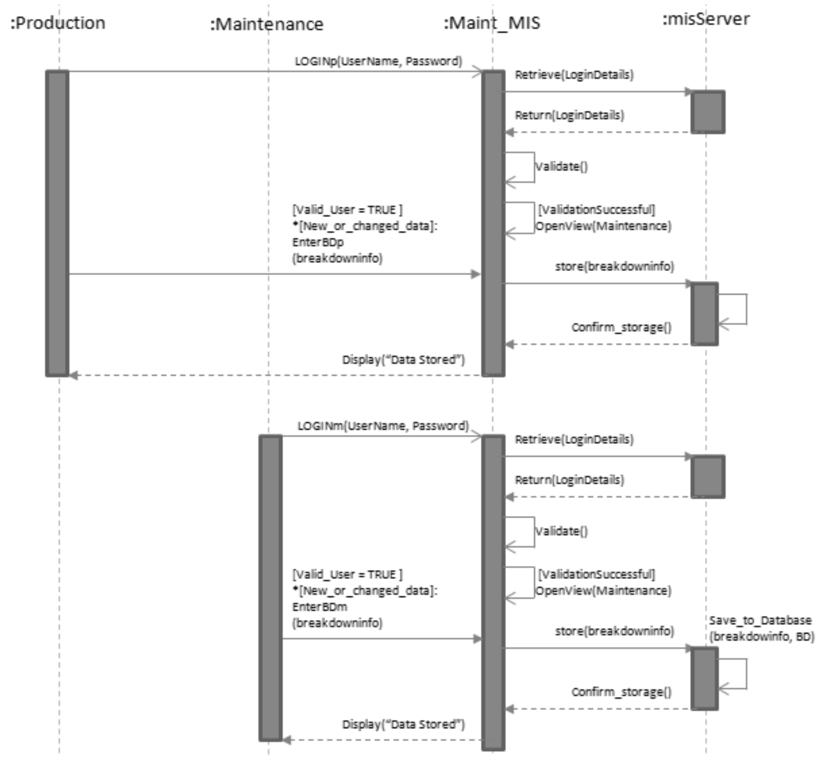

Figure 15. Sequence diagram 2

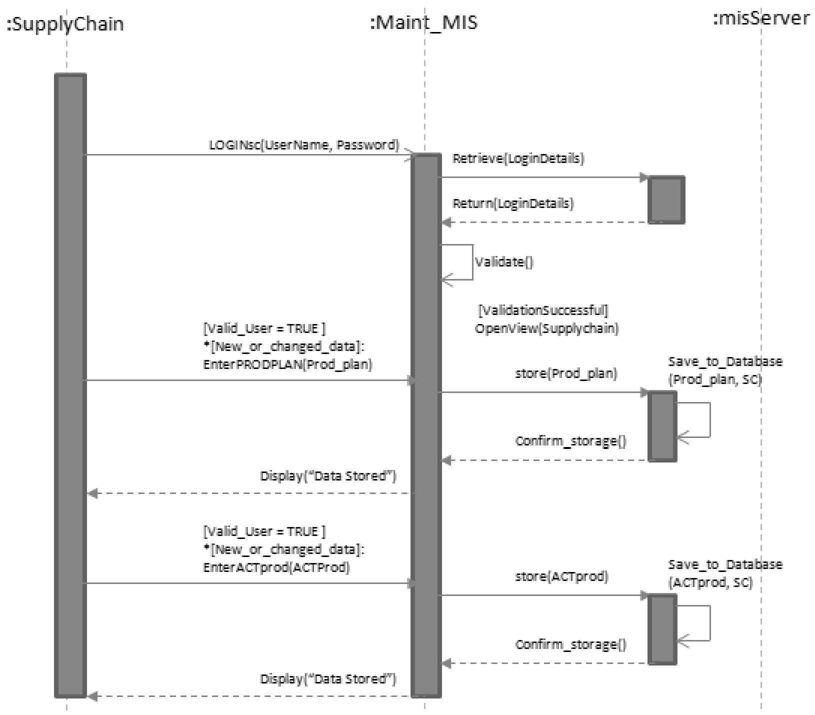

Figure 16. Sequence diagram 3

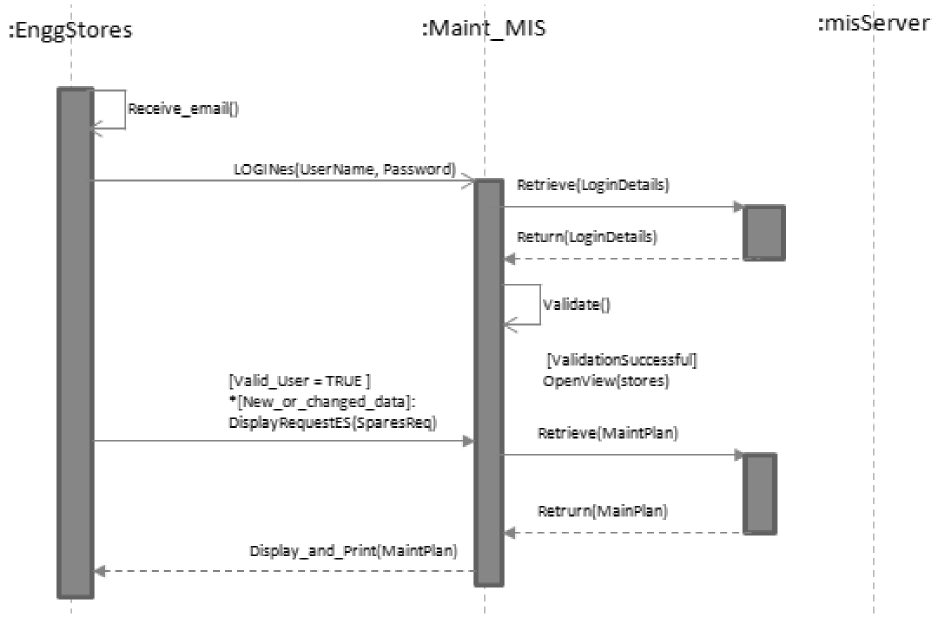

Figure 17. Sequence diagram 4

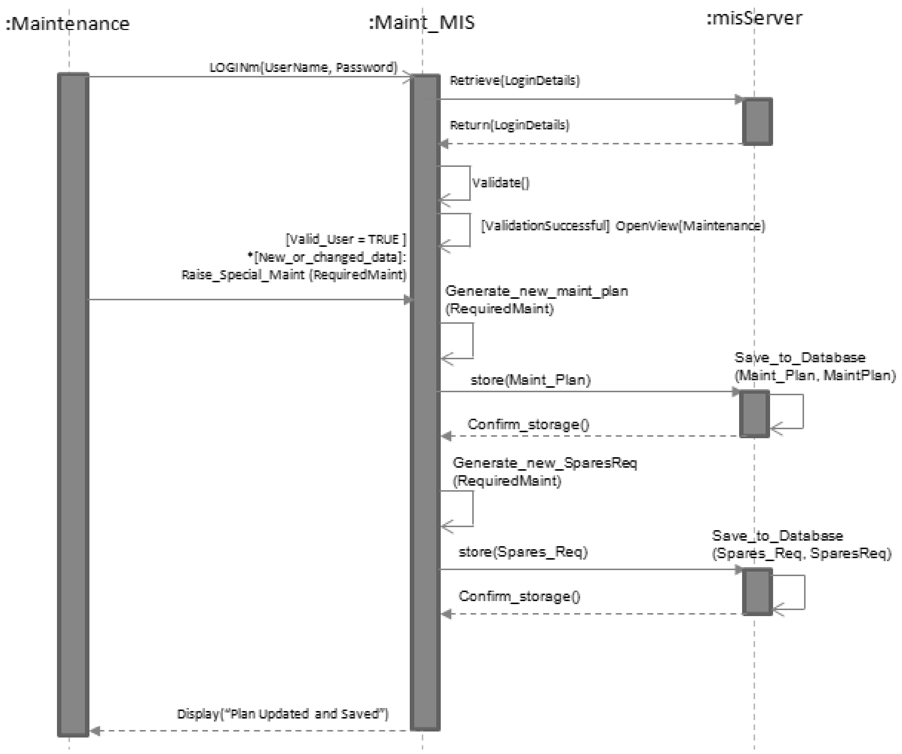

Figure 18. Sequence diagram 5

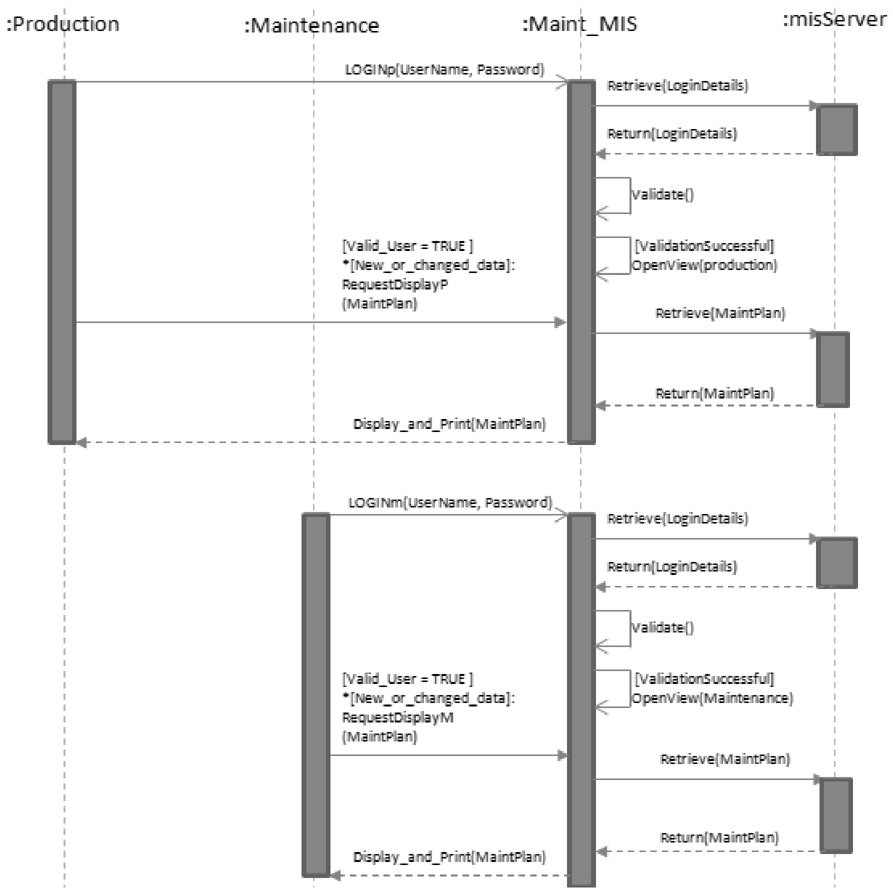

Figure 19. Sequence diagram 6

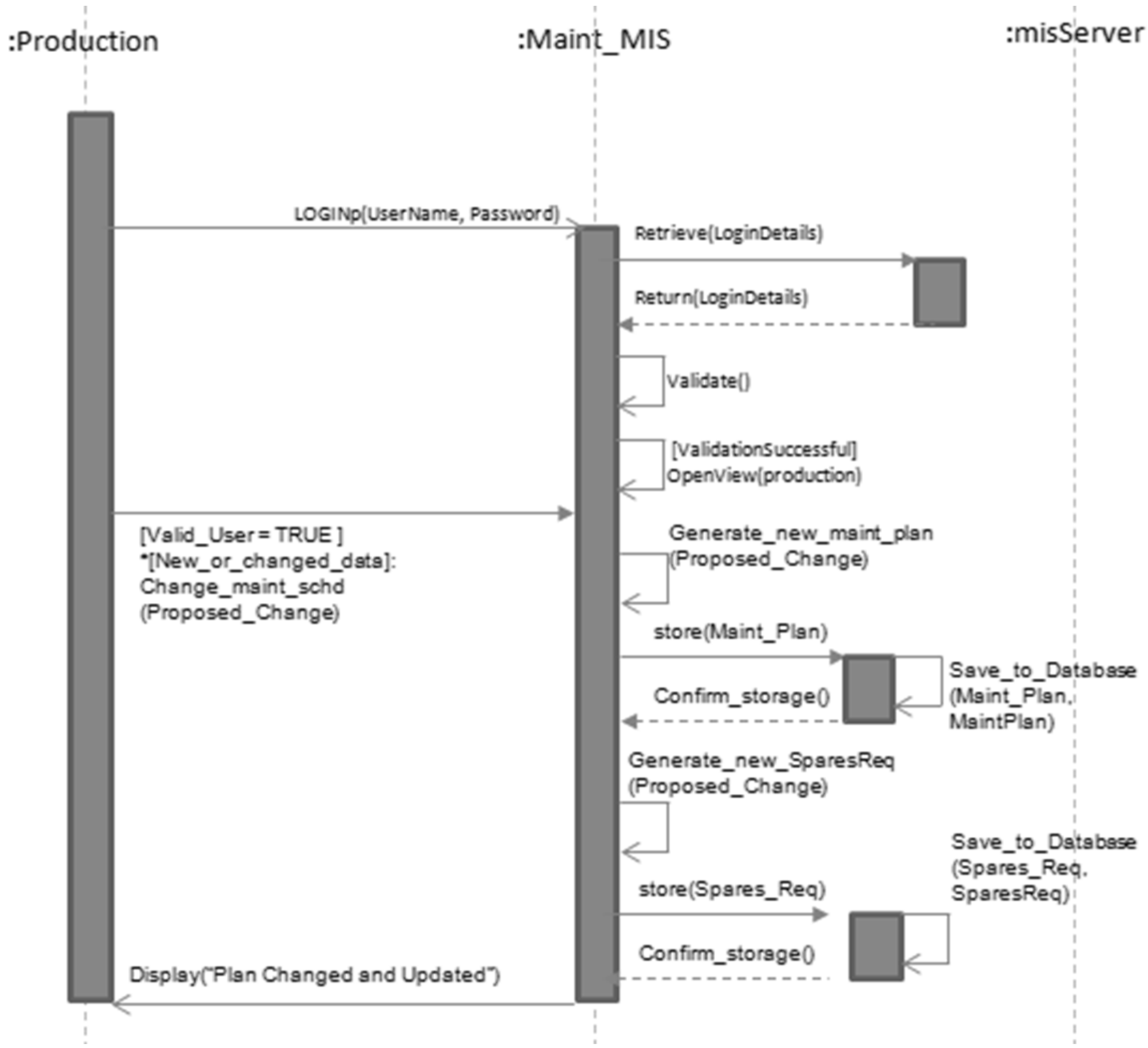

Figure 20. Sequence diagram 7

Reporting is one of the major functions of MIS. Accordingly, it involves a lot of relevant processes and sharing of information from one element of the system and organization to the other. Relative to other types of specialized information systems, an MIS is used by mid-level management to support ongoing operations. The emphasis is on making routine decisions. MIS relies mostly on internal sources of information (Zandbergen, 2015).

One of the important roles of an MIS is to provide the right information to the right person in the right format at the right time. Information is collected within the organization on an ongoing basis and an MIS processes this information, so managers get the summarized reports. Information is typically in the form of reports on a daily or weekly basis.

In our case, the next sequence diagram explains the reporting processes and how do different stakeholders deal with it.

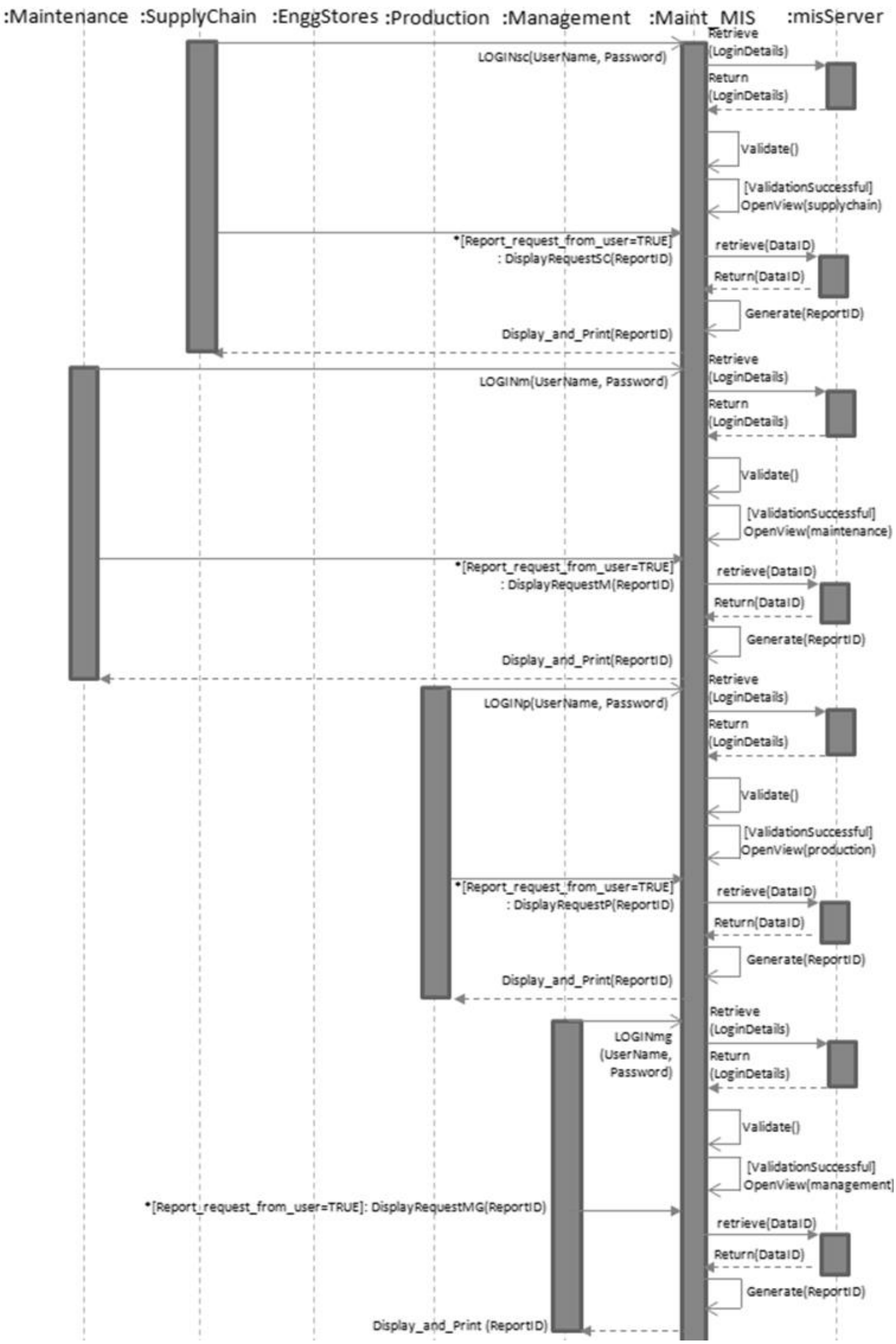

Figure 21: Sequence diagram for reporting

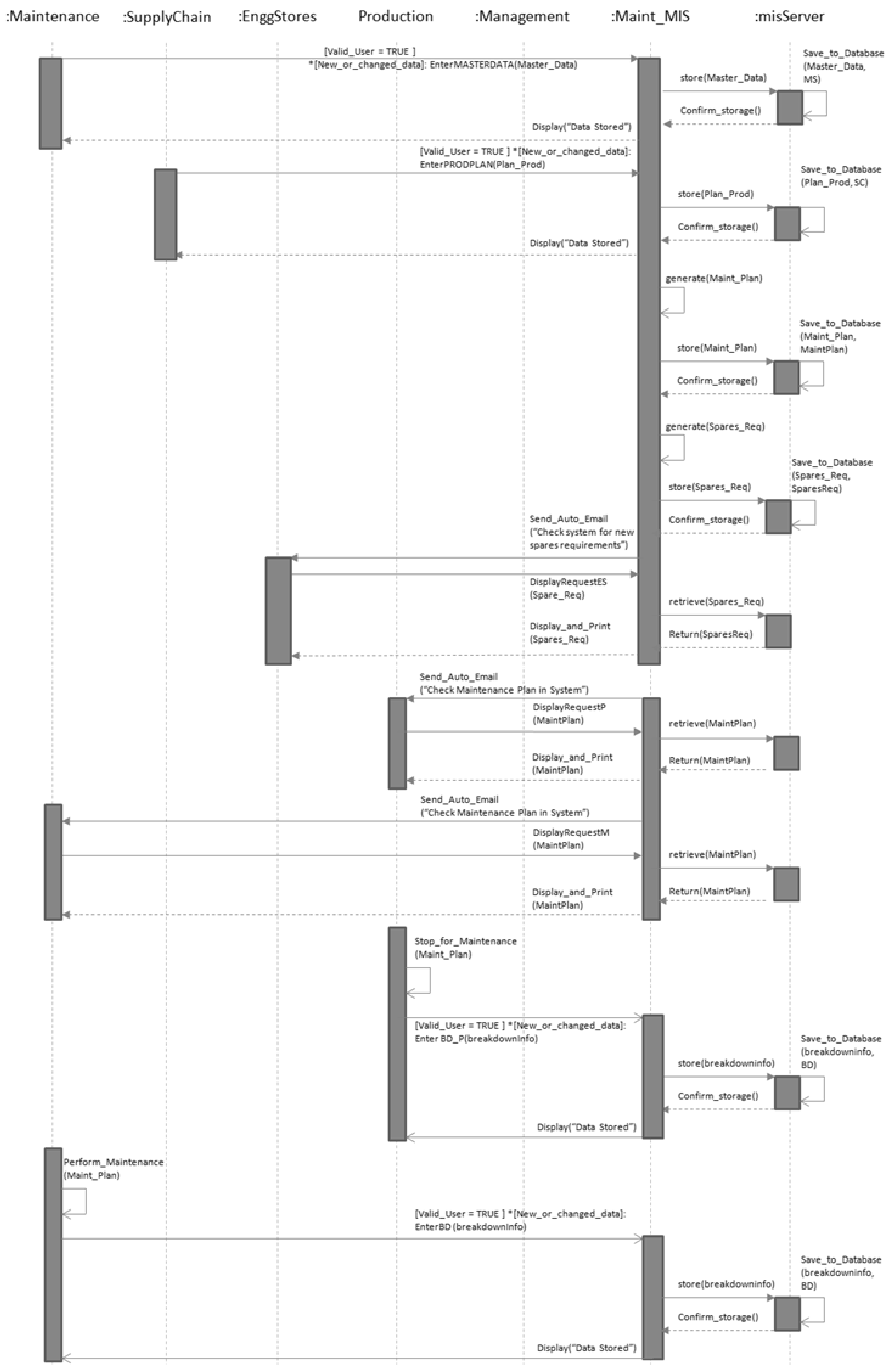

Figure 22: Overall sequence diagram part 1

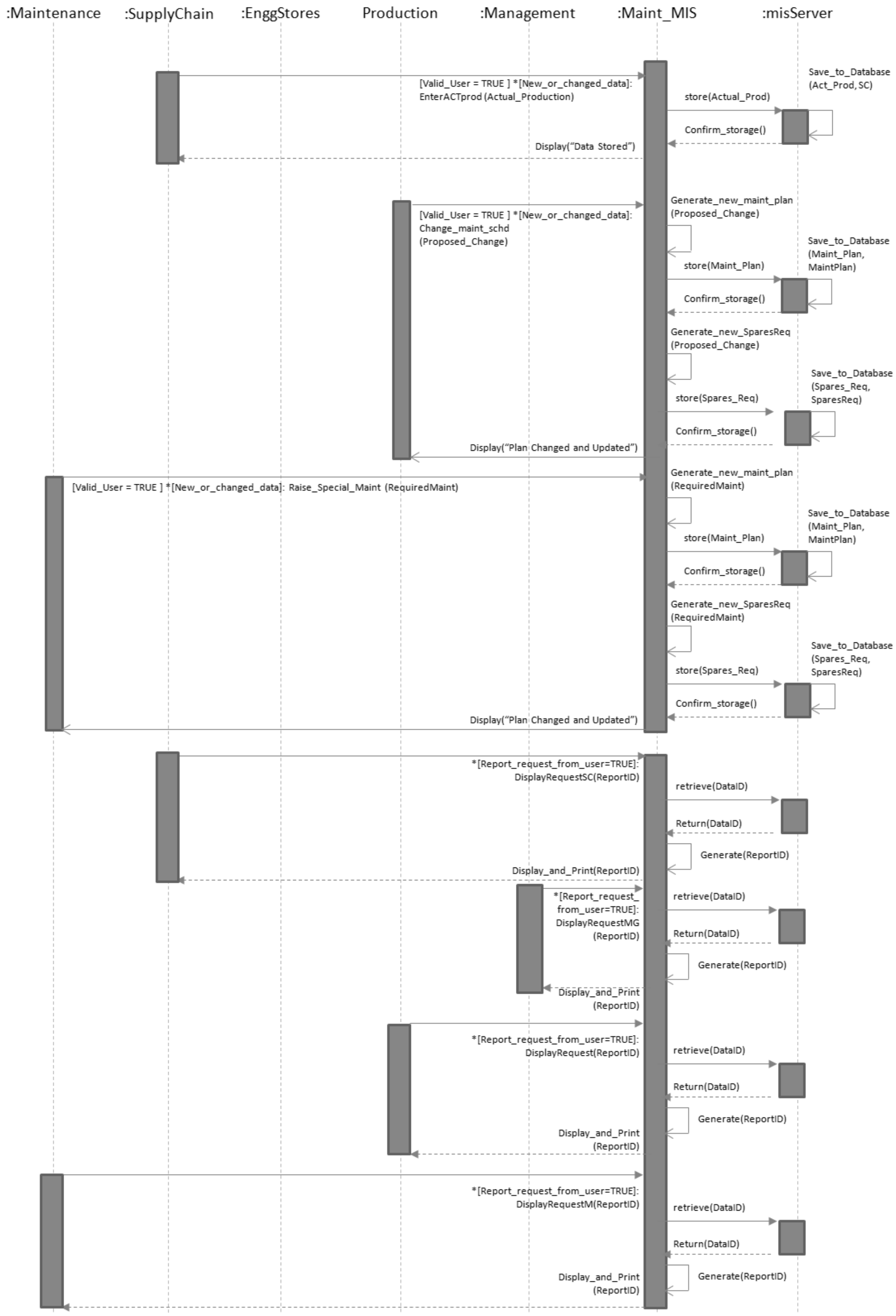

Figure 23: Overall sequence diagram part 2

ERD - ENTITY RELATIONSHIP DIAGRAMS:

An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and can be used as the foundation for a relational database.

Three main components of an ERD are the entities, which are objects or concepts that can have data stored about them, the relationship between those entities, and the cardinality, which defines that relationship in terms of numbers. The three main cardinal relationships are:

- One-to-one (1:1). For example, if each customer in a database is associated with one mailing address.
- One-to-many (1:M). For example, a single customer might place an order for multiple products. The customer is associated with multiple entities, but all those entities have a single connection back to the same customer.
- Many-to-many (M:N). For example, at a company where all call center agents work with multiple customers, each agent is associated with multiple customers, and multiple customers might also be associated with multiple agents.

Following are the ERDs for the project.

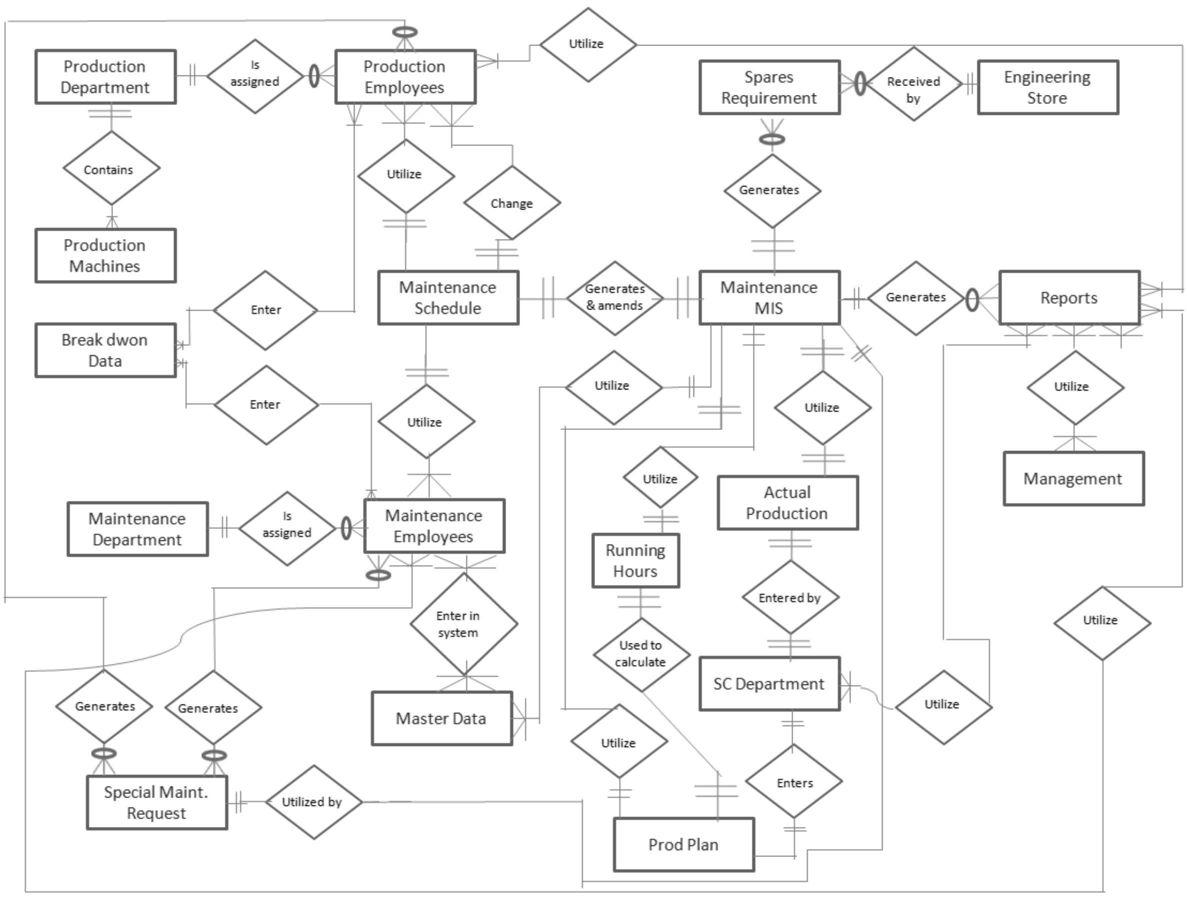

Figure 23: Entity relationship diagram

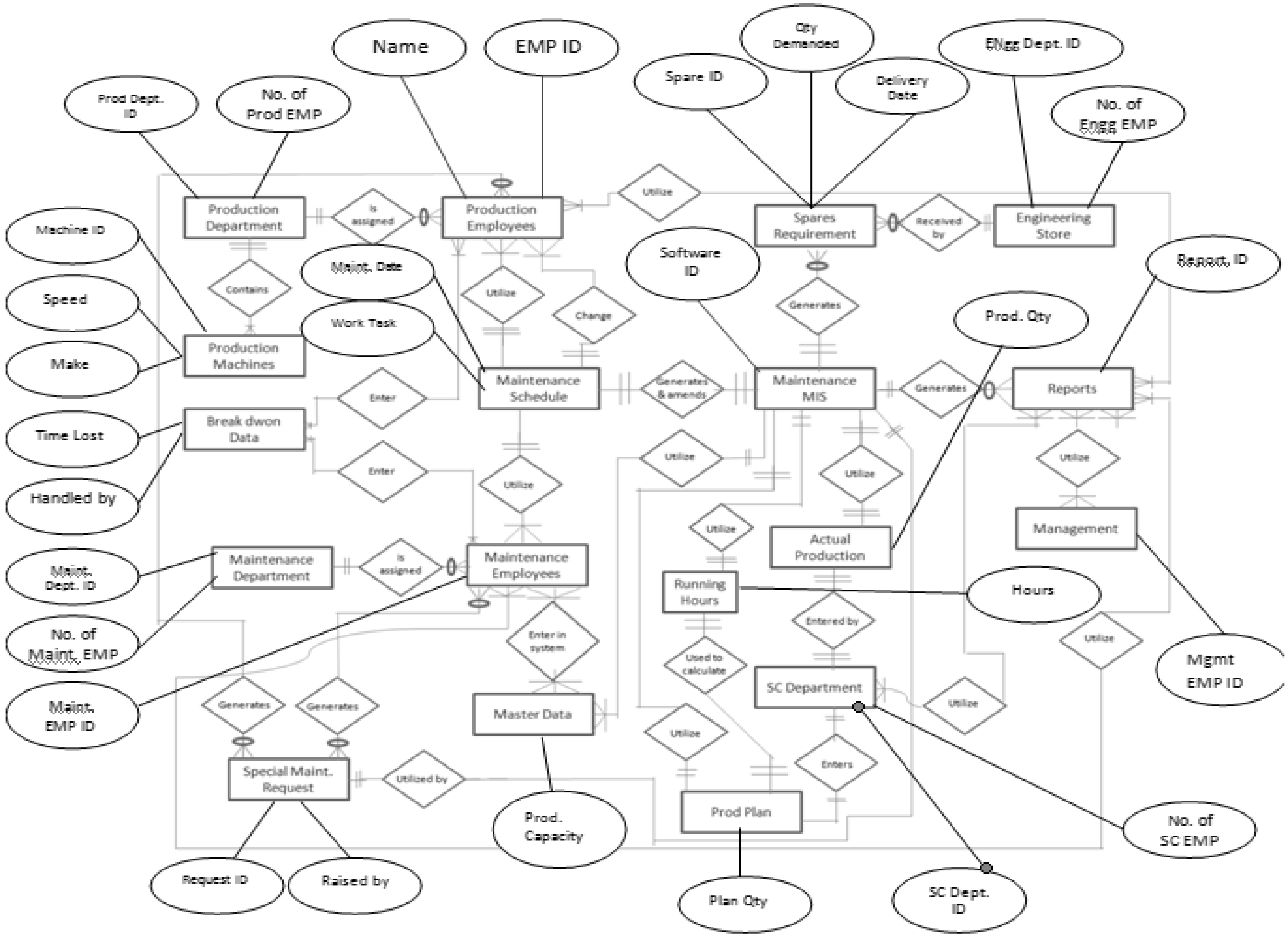

Figure 24: Elaborated Entity relationship diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

In the diagram, classes are represented by boxes that contain three compartments:

The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.

To avoid graphical congestion, CLASS constituents are not mentioned in the following the figure. However, they have elaborated in the next.

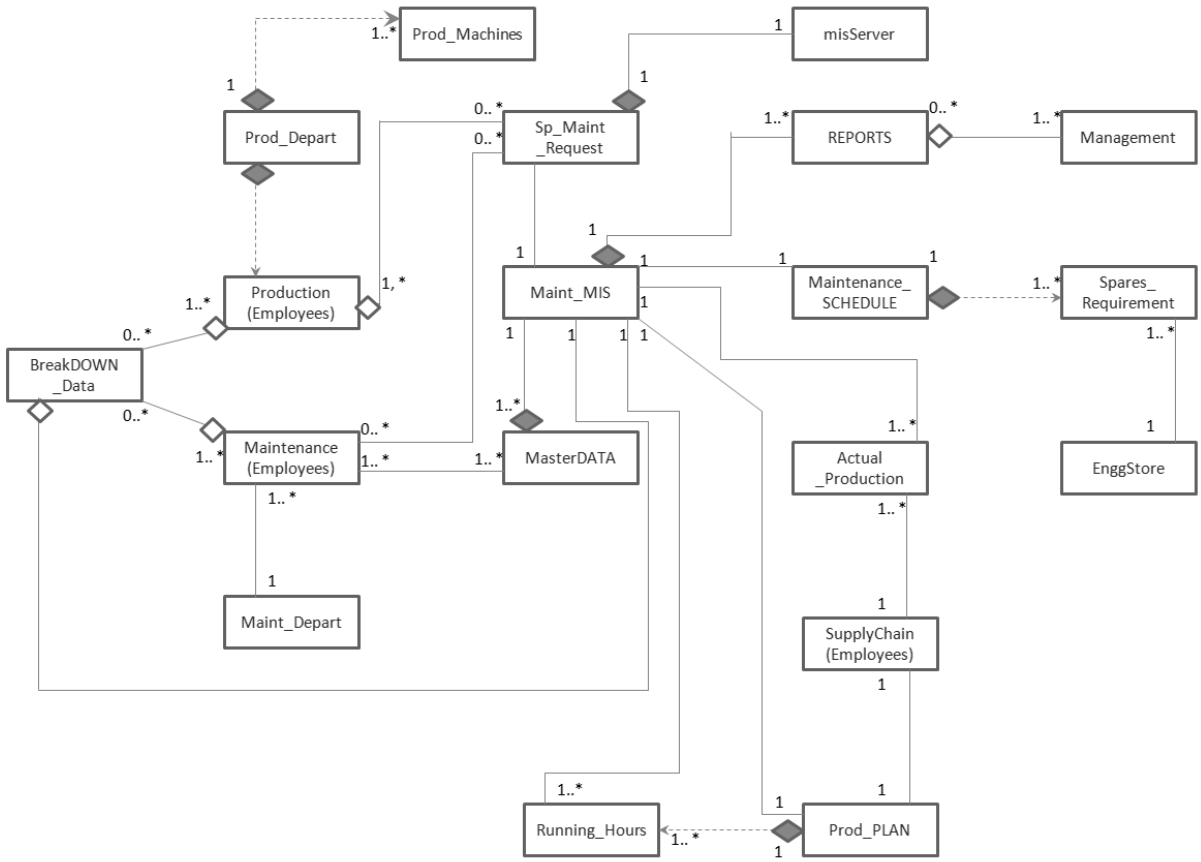

Figure 25: Broad overview class diagram

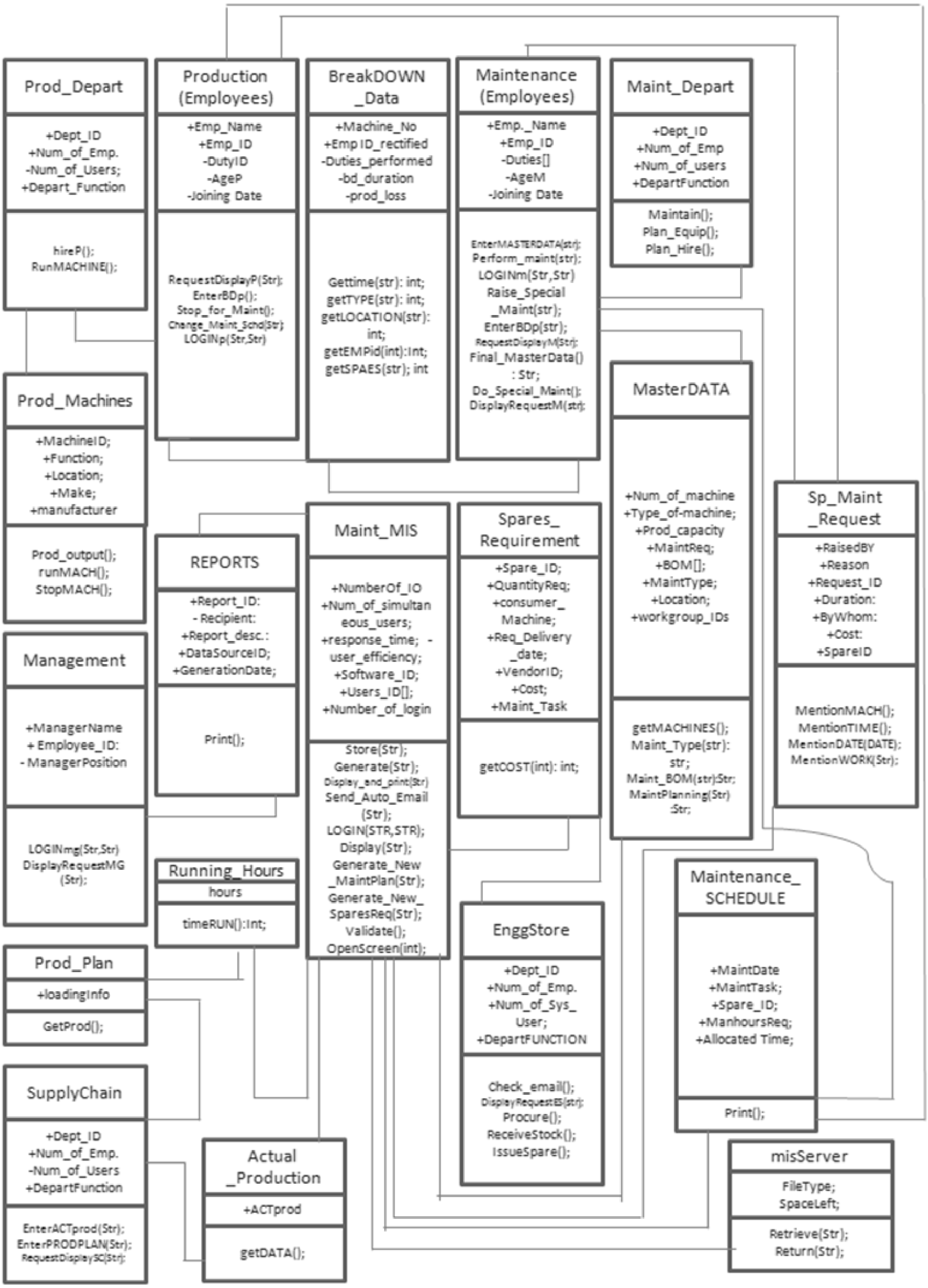

Figure 26: Detailed Class Diagram

DATABASE DIAGRAM:

A database diagram is a type of data model that determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized, and manipulated.

We can see that some data is required for more than 1 function. In such situation, if system collects two sets of data, it would be catastrophic. Accordingly, in database diagram, it is mapped and accordingly planned in terms of organization and storage.

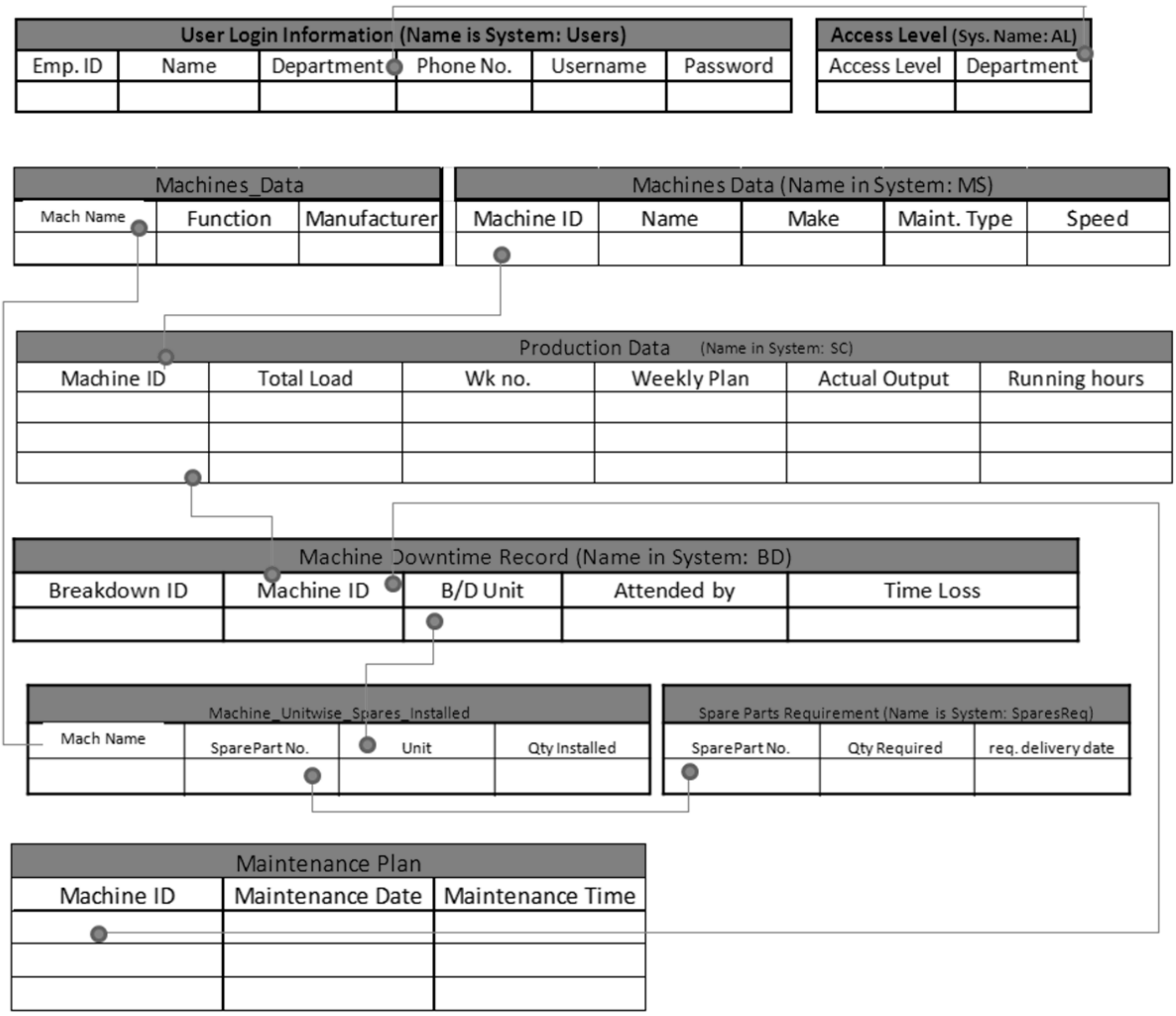

Figure 27: Database Diagram

TEST CASES:

A test case has components that describe an input, action or event, and an expected response, to determine if a feature of an application is working correctly ("How to write effective Test cases?," 2016). Writing effective test cases is a skill and that can be achieved by some experience and in-depth study of the application on which test cases are being written.

Following are test-cases for the current project.

Test Cases	
Title	Testing the Login Mechanism of Maintenance MIS System
Actions	Create user login Go to the Login screen Enter Login Name and Password Press Proceed Button
Expected Results	System Checks if the user is valid or not; If the user is valid, system logs in
Tested by	*To be updated at the time of test*
Result	*To be updated after the test*
Title	View System Generated Reports
Actions	Login to the system From the home screen, select 'View Reports' Select the required report Click Ok
Expected Results	System Checks if the user is valid or not; If the user is valid, system logs in; If above mentioned steps are performed correctly, requested report is shown on the screen
Tested by	*To be updated at the time of test*
Result	*To be updated after the test*
Title	*Enter Master Data into the system*
Actions	*Login to the system* *From the home screen, select 'Master Data'* *Select the required entry option from master data if modifying existing master data or filling pre-selected attributes* *If new master data element is to be made, select 'new master data' and enter the data in the relevant fields* *Click Ok*
Expected Results	*System Checks if the user is valid or not; If the user is valid, system logs in; System will check the access level; If the person logged in is authorized to build/ modify master data, system proceeds, otherwise, error message appears and software returns back to home screen*
Tested by	*To be updated at the time of test*
Result	*To be updated after the test*
Title	*Enter Production Plan in the system*
Actions	*Login to the system* *From the home screen, select 'SC Module'* *Select the required 'Production Plan' entry option from the list.*

	Enter the data in the relevant fields *Click Ok*
Expected Results	*System Checks if the user is valid or not; If the user is valid, system logs in; System will check the access level; If the person logged in is authorized to work in supply chain module, system proceeds, otherwise, error message appears and software returns back to home screen*
Tested by	*To be updated at the time of test*
Result	*To be updated after the test*
Title	*Checking Spare Part demand from the system*
Actions	*Login to the system* *From the home screen, select 'Engineering'* *Select the required 'Spares Demand' option from the list.* *Select from the given options including 'urgently required', 'for making machines', 'for making machines' or 'all requirements'* *Click Ok*
Expected Results	*System Checks if the user is valid or not; If the user is valid, system logs in; System will check the access level; If the person logged in is authorized to work in Engineering module, system proceeds, otherwise, error message appears and software returns back to home screen*
Tested by	*To be updated at the time of test*
Result	*To be updated after the test*
Title	*Checking Maintenance Plan from the System*
Actions	*Login to the system* *From the home screen, select 'Maintenance Schedule'* *Select from the given options including 'One week', 'fortnightly', 'monthly', 'Special maintenance' or 'all'* *Click Ok*
Expected Results	*System Checks if the user is valid or not; If the user is valid, system logs in;* *If type of maintenance plan view required is not picked from the list, system gives an error message and requests for selection;* *If correct selection is made, system shows the plan on screen*
Tested by	*To be updated at the time of test*
Result	*To be updated after the test*
Title	*Change Maintenance Plan*
Actions	*Login to the system* *From the home screen, select 'Plant Maintenance' Module* *Select from the given options 'Change Maintenance Schedule'* *Enter relevant data in the form opened* *Click Ok*
Expected Results	*System Checks if the user is valid or not; If the user is valid, system logs in;*

	The system checks the access level and if found ok, proceeds with the data entry. *When the submit button is pressed, system generates new maintenance plan and notifies the stakeholders about the change*
Tested by	*To be updated at the time of test*
Result	*To be updated after the test*
Title	*Raise Special Maintenance Request*
Actions	*Login to the system* *From the home screen, select 'Plant Maintenance' Module* *Select from the given options 'Special Maintenance'* *Enter relevant data in the form opened* *Click Ok*
Expected Results	*System Checks if the user is valid or not; If the user is valid, system logs in;* *The system checks the access level and if found ok, proceeds with the data entry.* *When the submit button is pressed, system generates new maintenance plan and notifies the stakeholders about the change*
Tested by	*To be updated at the time of test*
Result	*To be updated after the test*
Title	*Enter Breakdown Data*
Actions	*Login to the system* *From the home screen, select 'Plant Maintenance' Module* *Select from the given options 'Breakdown Maintenance'* *Enter relevant data in the form opened* *Click Ok*
Expected Results	*System Checks if the user is valid or not; If the user is valid, system logs in;* *The system checks the access level and if found ok, proceeds with the data entry.* *When the submit button is pressed, system shows analysis of the breakdown and recovery process*
Tested by	*To be updated at the time of test*
Result	*To be updated after the test*

CONCLUSION:

In this paper, we have presented the design of a simple but effective maintenance information management system with various phases during the design except software coding which would be a futile effort without customizing the model and phases according to specific organizational and company specific factors. The scope of functions can also be adjusted to include relevant requirements. The proposed model can also be used for base-level evaluation of existing Maintenance information systems and utilized for implementing continuous improvement initiatives. It is also to be noted that for the full ERP, a number of other aspects will need to be included.

REFERENCES:

Fountas, S., Sorensen, C. G., Tsiropoulos, Z., Cavalaris, C., Liakos, V., & Gemtos, T. (2015). Farm machinery management information system. *Computers and Electronics in Agriculture*, *110*, 131–138. http://doi.org/10.1016/j.compag.2014.11.011

How to write effective Test cases? Test cases procedures and definitions. (2016). Retrieved from http://www.softwaretestinghelp.com/how-to-write-effective-test-cases-test-cases-procedures-and-definitions/

Microsoft Corporation. (2015). Three-Layered Services Application. Retrieved April 16, 2016, from https://msdn.microsoft.com/en-us/library/ff648105.aspx

Use Cases. (2013, October 9). Retrieved April 16, 2016, from http://www.usability.gov/how-to-and-tools/methods/use-cases.html

Website Requirements. (2013, September 26). Retrieved April 16, 2016, from http://www.usability.gov/how-to-and-tools/methods/requirements.html

Zandbergen, P. (2015). Management Information Systems (MIS): Manager Decision-Making Tools - Video & Lesson Transcript. Retrieved April 16, 2016, from http://study.com/academy/lesson/management-information-systems-mis-manager-decision-making-tools.html